A basic problem of resource allocation in target tracking

Harilaos N. Psaraftis and Anastassios N. Perakis

Department of Ocean Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139

Stochastic dynamic programming techniques are used to formulate and solve the problem of tracking two independent and stationary targets with one sensor in order to maximize a certain measure of performance. At any point in time, the sensor, usually a passive sonar array, can be allocated to only one of the two targets. Assuming the fluctuation process in the ocean to be governed by a phase-random multipath law, the sensor "holds" the target when p, the root-mean-square pressure at the receiver, is above a user-specified threshold. Using discrete time models for the ocean acoustic detection process formulated in earlier papers, we solve the problem for a finite horizon of observations using several alternative objective and reward/penalty functions. Delays of user-specified magnitude in "switching" from one target to the other are also incorporated in our algorithms. Examples using both real and simulated data are presented and discussed. Finally, future research directions are suggested.

PACS numbers: 43.60.Gk, 43.30.Vh

INTRODUCTION: THE PROBLEM

Consider the following oversimplified situation in passive underwater acoustic surveillance: we are faced with the problem of tracking two stationary and independent targets with one sensor so as to maximize a certain measure of performance, which will be shortly defined. At any given point in time, the sensor, which can be for instance a passive sonar array, can be allocated only to one of the two targets for some exogeneous reason that is of no concern here. If at that point in time the root-mean-square pressure p of the signal due to that target is above a specified detection threshold p_0, we say that the target is "held" by the sensor. Holding a target is generally assumed to be a desirable outcome. However, due to the randomness of the fluctuation process, holding will not always occur. Assuming that the process is governed by a phase-random multipath law, there will be periods of time when the signal will be above the threshold and periods when it will be below it. At those times when the signal due to one target is below the threshold, it may make little or no sense keeping the sensor allocated to that target, especially if there is a certain chance of being able to hold the other target instead. The problem we will be trying to analyze and solve in this paper can be roughly phrased as follows: if we have some probabilistic information about *when* each target is likely to be above or below its detection threshold, is there an "optimal" allocation (or switching) schedule of the sensor between the two targets through time?

The above rather sketchy description classifies the problem at hand in the general category of "resource allocation" problems, which is of particular importance in target tracking and passive underwater acoustic surveillance.

Resource allocation generally calls for simultaneously tracking a number of targets with a limited number of sensors already deployed in a geographical area of interest. Limitations may be due to a number of constraints, such as number of available sensors, number of available communication channels, information processing capacity, bandwidth, etc. There is an abundance of literature in this area, each paper tackling a particular version of the problem, and

each using different methods that can be grouped into broad categories. For instance, Alspach[1] uses a Gaussian-sum approach to the multitarget identification-tracking problem, under the somewhat unrealistic assumptions of no missing measurements and no false measurements. Bar Shalom, in cooperation with various authors, has produced various papers in the area, using stochastic dynamic programming for resource allocation under uncertainty[2] and presenting target tracking algorithms with uncertain detection origins and random "interdetection" times.[3] In the latter paper, suboptimal versions are presented and probabilistic data association filters (PDAF) are employed to associate the measurements to the corresponding target. Noise is assumed zero mean, white, and Gaussian and the problem is in the Kalman filtering mold (linear state and observation equations). A 1978 survey paper by Bar Shalom[4] presents a comprehensive discussion of existing tracking methods in a multitarget environment, dividing them into two broad groups: Bayesian (using *a posteriori* probabilities) and non-Bayesian (using likelihood functions) and extensively presenting their pros and cons. A qualitative paper on problems in multitarget sonar tracking[5] is concerned with the shortcomings of the extended Kalman filter (EKF) and probabilistic decision making in adaptive tracking for ocean surveillance, also outlining several avenues for future research. A group from Bolt Beranek and Newman (BBN) has carried out some research on the approximate evaluation of the covariance matrices associated with target state estimates.[6,7] In Friedlander,[8] observations are modeled as a multichannel auto regressive moving average (ARMA) process. Hypothesis testing involving pruning and merging of hypotheses is employed in the algorithms of Keverian and Sandell.[9] A similar approach is used in Reid.[10] A common shortcoming of all the above papers and of others not mentioned here is that no one has comparisons of its theoretical models with real data—at most simulation results are presented.

The problem examined in this paper does not attempt to get involved with the intricate signal processing and tracking issues associated with the real world environment. This version of the problem is rather trivial (two targets, one sensor)

and unrealistic (stationary targets, sensor can be allocated only to one target). Rather, the rationale behind this approach is to take advantage of recent results in the modeling of the ocean acoustic detection process.[11,12] These results have shed some light into the timing of detection events in the ocean. Specifically, it has been shown for the case of a phase random multipath acoustic process—adequately describing long range acoustic signal propagation—that the detection process has memory, and that information about the frequency of occurrence of detections and the duration of the "interarrival" and "holding" intervals can be readily obtained. This paper investigates how one can take advantage of the dynamics of the detection process for a more efficient allocation of resources, beginning by the examination of the (least complicated) two target–one sensor problem. The algorithm developed for this problem is subsequently readily extended to take into account possible delays in tracking of a target to which the sensor has just been allocated. The original version of the problem assumes that if switching between targets occurs, the sensor has the capability of instantaneously "tuning" to the new target and hence determining whether or not that target is held. However, this may not be realistic. The "statistic" p which is tested against the detection threshold p_0 is usually the output of some integration processing of the actual real-time fluctuations received by the sensor. The fact that this integration process takes time implies that there may be in general an interval after each switching during which we will have no information on whether we hold the target or not.

In the remainder of this paper, Sec. I formulates the original version of the problem and solves it using dynamic programming. Several alternative objective functions concerning rewards/penalties for detecting/holding/losing a target are considered and discussed. Section II extends the above formulation by incorporating delays into the problem.

Section III presents some examples so as to illustrate the algorithms developed. Finally, Sec. IV discusses the results of this work and suggests directions for further research.

I. PROBLEM FORMULATION AND SOLUTION

Previous work[12] has suggested that for a phase random multipath acoustic process, fluctuations from each target can be modeled as a discrete-time, two-state Markov process. Referring to Fig. 1, states labeled U (for "up") imply that fluctuations due to the target in question are above the detection threshold, and states labeled D (for "down") imply the opposite. The transition probabilities (a_i, b_i), are assumed to be known for each target $i (i = 1,2)$. Opportunities for a state transition occur simultaneously for each target every ΔT units of time. The reader is referred to Ref. 12 for details on how the transition probabilities can be calculated from the detection threshold and other parameters, as well as on how ΔT can be calibrated. For the purposes of this paper the above parameters will be assumed known and constant through time.

Our sensor can be tuned to either one of the two targets (but not to both) at any point in time. If it is tuned to, say, target 1, the state of that target is known $(U_1$ or $D_1)$. Of course during that time, the state of target 2 is not known with certainty. We assume that every ΔT units of time the sensor has the opportunity to switch to the other target, or continue being tuned to the original one.

At any discrete point in time the state of the system can be described by the triplet (i, j, p) with i being the target at which the sensor is tuned to $(i = 1,2)$, j being the state of that target $(j = 1$ for U and $j = 2$ for D $)$, and p being the probability that at that point in time the state of the other target is U.

Assuming that the state of the system at some discrete point in time, say, $n\Delta T$, is (i, j, p), then, depending on whether we choose to tune to target 1 or 2, the state of the
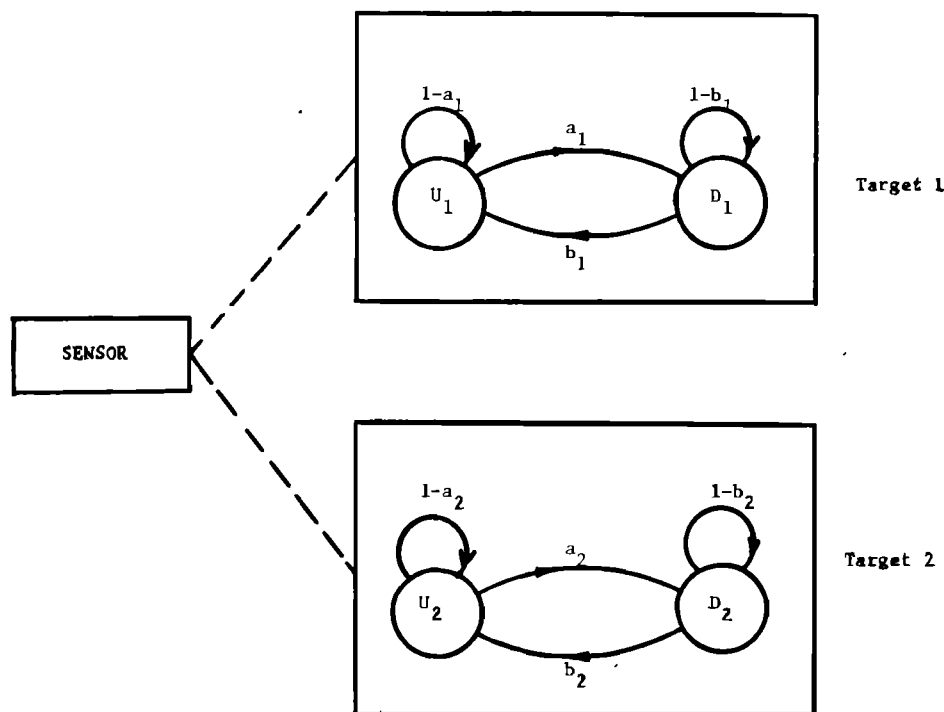

FIG. 1. Discrete-time Markov models for targets.

TABLE I. Three possible forms of [c].

(i',j') (1,1)	(1,2)	2,1)	(2,2)		(i',j') (1,1)	(1,2)	(2,1)	(2,2)		(i',j') (1,1)	(1,2)	(2,1)	(2,2)
(i,j)					(i,j)					(i,j)			
(1,1) 1	0	1	0		(1,1) 1	0	1	0		(1,1) 1	0	1	0
(1,2) 1	0	1	0		(1,2) 0	1	0	0		(1,2) 0	1	0	-1
(2,1) 1	0	1	0		(2,1) 1	0	1	0		(2,1) 1	0	1	0
(2,2) 1	0	1	0		(2,2) 0	0	0	1		(2,2) 0	-1	0	1
(a)					(b)					(c)			

system at the next discrete point in time $(n + 1)\Delta T$ will be $(i',j',p',)$ with i' being 1 or 2 accordingly. j' cannot be known with certainty prior to making that decision. In the following, the probability that j' is 1 or 2 will be shown to depend on i,j,i',p, and the transition probabilities. Similarly, p' will be shown to depend on the values of i,j,i',j',p, and the transition probabilities.

Given the values of i and j, not all possible outcomes (i',j') will be, in general, equally desirable. For instance, given $(i,j) = (1,1)$ (target 1 is held), outcome $(i',j') = (1,1)$ (target 1 continues to be held) will be more desirable than outcome $(i',j') = (2,2)$ (we switch to target 2 but we cannot hold it). Or given $(i,j) = (1,2)$ (target 1 is lost), outcome $(i',j') = (2,1)$ (we switch to target 2 and we hold it) will be more desirable than
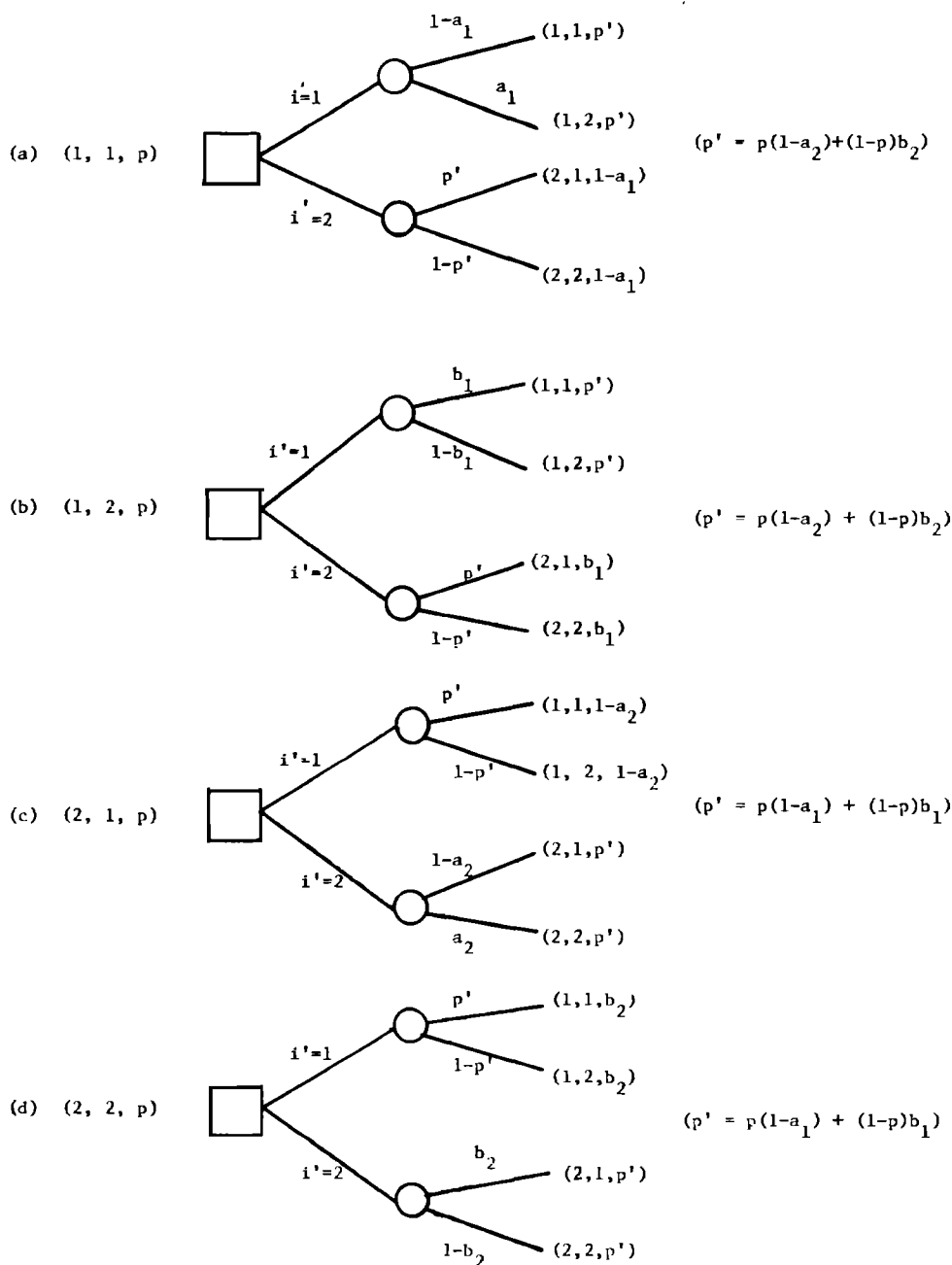
(a) (1, 1, p)

$i'=1$ → $1-a_1$ → $(1,1,p')$
$i'=1$ → a_1 → $(1,2,p')$
$i'=2$ → p' → $(2,1,1-a_1)$
$i'=2$ → $1-p'$ → $(2,2,1-a_1)$

$(p' = p(1-a_2)+(1-p)b_2)$

(b) (1, 2, p)

$i'=1$ → b_1 → $(1,1,p')$
$i'=1$ → $1-b_1$ → $(1,2,p')$
$i'=2$ → p' → $(2,1,b_1)$
$i'=2$ → $1-p'$ → $(2,2,b_1)$

$(p' = p(1-a_2) + (1-p)b_2)$

FIG. 2. State transitions in the DP algorithm.

(c) (2, 1, p)

$i'=1$ → p' → $(1,1,1-a_2)$
$i'=1$ → $1-p'$ → $(1, 2, 1-a_2)$
$i'=2$ → $1-a_2$ → $(2,1,p')$
$i'=2$ → a_2 → $(2,2,p')$

$(p' = p(1-a_1) + (1-p)b_1)$

(d) (2, 2, p)

$i'=1$ → p' → $(1,1,b_2)$
$i'=1$ → $1-p'$ → $(1,2,b_2)$
$i'=2$ → b_2 → $(2,1,p')$
$i'=2$ → $1-b_2$ → $(2,2,p')$

$(p' = p(1-a_1) + (1-p)b_1)$

outcome $(i',j') = (1,2)$ (we do not switch but still target 1 remains lost).

We assume a general reward/penalty function $c(i,j,i',j')$ for quantifying how desirable each outcome is. The 16-element matrix $[c]$ is assumed to be a user input, with its elements being positive, negative, or zero. Three possible forms of $[c]$ are displayed in Table I: in (a), we are rewarded by one unit each time we hold either one of the targets independently of whether we were holding a target at the previous stage or not. In (b), we are rewarded by one unit each time we record an upcrossing or a downcrossing. The rationale here is that we consider any information about when those events occur to be very important, without rewarding our being in state "U" more than being in "D." Matrix (b) also assumes that we are rewarded with one unit when we do not hold a target at stage n but hold a target if we switch at the

next stage, hence without necessarily having a crossing. In (c), we have the same structure as in (b), yet we are penalized (with a negative unit) if we do not hold a target at stage $n+1$ while we were holding one (not necessarily the same one) at stage n.

The spectrum of all possible state transitions (i,j,p) to (i',j',p') along with their associated probabilities is displayed in Fig. 2 (chance nodes are circles and decision nodes are squares). Figure 2 also displays the relationship of p' with i,j, i',j', and p.

We are now in a position to develop a recursive relationship to solve our problem. Let $V_n(i,j,p)$ be the maximum expected total net reward from stage n to the end N of the process, given at stage n the state of the system is (i,j,p). Taking Fig. 2 into account we have

(a) $0 \leqslant n < N$, $0 \leqslant p \leqslant 1$

$$V_n(1,1,p) = \max\begin{cases}(1-a_1)[c(1,1,1,1) + V_{n+1}(1,1,p')] + a_1[c(1,1,1,2) + V_{n+1}(1,2,p')], \\ p'[c(1,1,2,1) + V_{n+1}(2,1,1-a_1)] + (1-p')[c(1,1,2,2) + V_{n+1}(2,2,1-a_1)]\end{cases} \quad (1)$$

$$V_n(1,2,p) = \max\begin{cases}b_1[c(1,2,1,1) + V_{n+1}(1,1,p')] + (1-b_1)[c(1,2,1,2) + V_{n+1}(1,2,p')], \\ p'[c(1,2,2,1) + V_{n+1}(2,1,b_1)] + (1-p')[c(1,2,2,2) + V_{n+1}(2,2,b_1)]\end{cases} \quad (2)$$

$$V_n(2,1,p) = \max\begin{cases}p'[c(2,1,1,1) + V_{n+1}(1,1,1-a_2)] + (1-p')[c(2,1,1,1) + V_{n+1}(1,2,1-a_2)], \\ (1-a_2)[c(2,1,2,1) + V_{n+1}(1,1,p')] + a_2[c(2,1,2,2) + V_{n+1}(1,2,p')]\end{cases} \quad (3)$$

$$V_n(2,2,p) = \max\begin{cases}p'[c(2,2,1,1) + V_{n+1}(1,1,b_2)] + (1-p')[c(2,2,1,2) + V_{n+1}(1,2,b_2)], \\ b_2[c(2,2,2,1) + V_{n+1}(2,1,p')] + (1-b_2)[c(2,2,2,2) + V_{n+1}(2,2,p')]\end{cases}. \quad (4)$$

(b) $n = N$, $0 \leqslant p \leqslant 1$

Then $V_n(i,j,p) = 0$, for $i = 1,2$, $j = 1,2$, and $0 \leqslant p \leqslant 1$.

If, instead of maximizing the expected reward for the remaining stages, we are interested in obtaining the best possible "worst case" performance, we should maximize the minimum anticipated reward of the remaining stages (for each of the two available decisions). The recursion will assume the following form:

$$V_n(1,1,p) = \max\begin{cases}\min[c(1,1,1,1) + V_{n+1}(1,1,p'), \ c(1,1,1,2) + V_{n+1}(1,2,p')], \\ \min[c(1,1,2,1) + V_{n+1}(2,1,1-a_1), \ c(1,1,2,2) + V_{n+1}(2,2,1-a_1)]\end{cases} \quad (5)$$

$$V_n(1,2,p) = \max\begin{cases}\min[c(1,2,1,1) + V_{n+1}(1,1,p'), \ c(1,2,1,2) + V_{n+1}(1,2,p')], \\ \min[c(1,2,2,1) + V_{n+1}(2,1,b_1), \ c(1,2,2,2) + V_{n+1}(2,2,b_1)]\end{cases} \quad (6)$$

$$V_n(2,1,p) = \max\begin{cases}\min[c(2,1,1,1) + V_{n+1}(1,1,1-a_2), \ c(2,1,1,2) + V_{n+1}(1,2,1-a_2)], \\ \min[c(2,1,2,1) + V_{n+1}(2,1,p'), \ c(2,1,2,2) + V_{n+1}(2,2,p')]\end{cases} \quad (7)$$

$$V_n(2,2,p) = \max\begin{cases}\min[c(2,2,1,1) + V_{n+1}(1,1,b_2), \ c(2,2,1,2) + V_{n+1}(1,2,b_2)], \\ \min[c(2,2,2,1) + V_{n+1}(2,1,p'), \ c(2,2,2,2) + V_{n+1}(2,2,p')]\end{cases}. \quad (8)$$

In addition to the above cost structure, which could be modified for the case of targets of unequal importance by having a different reward for holding target 2 than for target 1, we might also wish to consider the case where we are penalized if the value of the probability p that the other target is in state U is near 0.5, since this is the state of greatest uncertainty for that particular target. Our results in this paper are based on an additional term in the reward/penalty function, $-Fp(1-p)$, where the factor F can be any real number. {Some authors prefer $[p(1-p)]^{1/2}$ due to its nicer analytical properties in the calculation of error probabili-

ties.} This expression has a minimum of $p = 0.5$ and local maxima at $p = 0$ and 1.

II. DELAY MODELING

As mentioned in the Introduction, once we decide to switch from one target to the other, we may not necessarily be able to track it (tune to it, receive its ρ) for the next KD-1 stages. One reason for the above delays, that is, integration time greater than the time interval between two successive observations, was already discussed. Other reasons might

include slowness in human reaction, delays in the execution of the switching, particularly in the case of manual array steering, or, less possible, time lags in the transmission and execution of the switching decision if done automatically. In the following, KD will be a user-specified input to the algorithm, stage to stage. For KD-1 stages immediately following a switching and before we are able to tune to the new target, no rewards are incurred since no observations are available.

In fact, for $n = N$-KD,...,N and if the elements of the cost matrix are non-negative [$c(i,j,i',j')\geqslant 0$, $i,j,i',j' = 1,2$] there will be no further switchings, since we will incur no benefits by switching to the other target and hence we have to continue listening to the target currently tracked. For the remaining stages, and for the case of the maximization of the expected reward-to-go, the following algorithm will have to be used:

(a) $0 \leqslant n \leqslant N$, $0 \leqslant p \leqslant 1$

$$V_n(1,1,p) = \max \begin{cases} (1 - a_1)[c(1,1,1,1) + V_{n+1}(1,1,p')] + a_1[c(1,1,1,2) + V_{n+1}(1,2,p')], \\ g_2[c(1,1,2,1) + V_{n+KD}(2,1,g_1')] + (1 - g_2')[c(1,1,2,2) + V_{n+KD}(2,2,g_1')] \end{cases} \tag{9}$$

$$V_n(1,2,p) = \max \begin{cases} b_1[c(1,2,1,1) + V_{n+1}(1,1,p')] + (1 - b_1)[c(1,2,1,2) + V_{n+1}(1,2,p')], \\ g_2'[c(1,2,2,1) + V_{n+KD}(2,1,g_1'')] + (1 - g_2')[c(1,2,2,2) + V_{n+KD}(2,2,g_1'')] \end{cases} \tag{10}$$

$$V_n(2,1,p) = \max \begin{cases} g_3'[c(2,1,1,1) + V_{n+KD}(1,1,g_4')] + (1 - g_3')[c(2,1,1,2) + V_{n+KD}(1,2,g_4')], \\ (1 - a_2)[c(2,1,2,1) + V_{n+1}(1,1,p')] + a_2[c(2,1,2,2) + V_{n+1}(1,2,p')] \end{cases} \tag{11}$$

$$V_n(2,2,p) = \max \begin{cases} g_3'[c(2,2,1,1) + V_{n+KD}(1,1,g_4'')] + (1 - g_3')[c(2,2,1,2) + V_{n+KD}(1,2,g_4'')], \\ b_2[c(2,2,2,1) + V_{n+1}(2,1,p')] + (1 - b_2)[c(2,2,2,2) + V_{n+1}(2,2,p')] \end{cases} \tag{12}$$

(b) $n = N$, $0 \leqslant p \leqslant 1$

Then $V_N(i,j,p) = 0$, for $i = 1,2$, $j = 1,2$, and $0 \leqslant p \leqslant 1$.
In the above,

g_1' = prob(target 1 is up after KD steps|is down now),

g_1'' = prob(target 1 is up after KD steps|is up now),

g_2' = prob(target 2 is up after KD steps|is up with prob p_2 now),

g_3' = prob(target 1 is up after KD steps|is up with prob p_1 now),

g_4' = prob(target 2 is up after KD steps|is down now),

g_4'' = prob(target 2 is up after KD steps|is up now).

Analytical expressions for the g_i',g_i'''s as functions of KD can be easily derived from the transition probabilities of our two-state Markov models. They are of the general form

$$g = p\phi_{1i} + (1 - p)\phi_{2i}, \quad i = 1 \text{ or } 2,$$

where p = prob(target is in state "up") and the ϕ_{ji} are the KD-step transition probabilities of the Markov models, for example,

$$\phi_{12} = [a/(a + b)][1 - (1 - a - b)^{KD}],$$

where $a = p_{12}$ and $b = p_{21}$ in the Markov model of the target of interest.

III. RESULTS

The algorithms described in the previous two sections have been implemented using MIT's Joint Computer Facili-

ty (VAX/VMS). Discrete probability spaces (101 probabilities at 0.01 intervals) have been used, with horizons of at least $N = 15$ (for the denser probability grid) and up to 45 stages. Both real and simulated data were used from actual $p(t)$ records from the CASE experiment (described in Ref. 13) and the latter by Monte-Carlo methods using the corresponding Markov models. The Markov models were used throughout this research, due to their simplicity, quickness in obtaining results and most important, superior compatibility with discrete-time sequential algorithms over our continuous-time models.

We have developed a versatile program that, depending on the options chosen by the user, can employ delays of specified duration ($KD = 1$ means no delay), cost matrix [c] and factor F in $-Fp(1 - p)$. Real or simulated data can be used, taken at user-specified time intervals. These intervals, like KD and [c], are constant from stage to stage. However, the elements of [c] need not be constants; they could be known functions of the stage or even random variables, albeit with known—and finite—expected value.

The real data presented in this paper are from the CASE experiment (record 146 for target 1 and 132 for target 2). Simulated data were obtained by generating random numbers between 0 and 1 and comparing them with the probability of being "up" at each stage of Markov models corresponding to the same CASE records directly used for real data.

TABLE II. Characteristics of targets tracked.

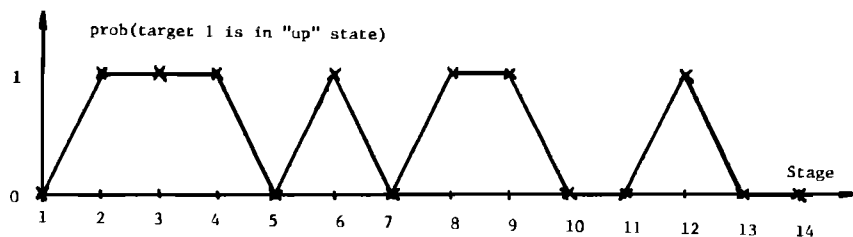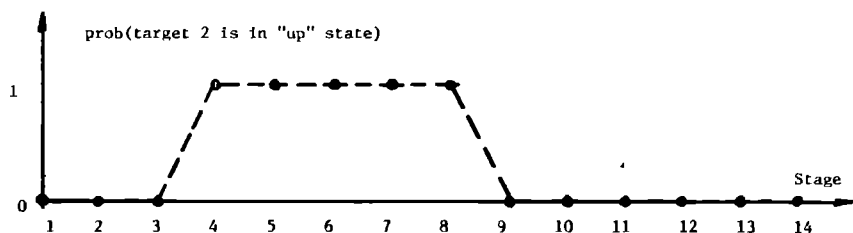| Target #	ρ_{0i} (V)	σ_{1i}^2(V^2)	v_i (Hz)	ΔT_i (s)	a_i	b_i	N
1	2.4	3.051	0.25419	0.72	0.51190	0.32603	15
2	1.6	1.291	0.13130	0.72	0.31760	0.18735	15

FIG. 3. Real data.

In Table II, ρ_{0j} are the detection threshold levels, σ_{ij}^2 the variances of the rms acoustic pressures, v_i the rms phase rates, ΔT_i the time steps of the relevant Markov models, a and b their p_{12} and p_{21}'s, respectively (see Fig. 1), and N the number of stages for the algorithm.

In the remaining figures of this report, we have put the stage numbers in the abscissas and the probability that our targets are in the "up" stage in the ordinates. The convention used is to depict target 1 with a cross "x" and target 2 with a circle "0."

Figures 3(a) and (b) present real data for the cases that we are tuned to targets 1 and 2 for the entire duration of the observations, respectively. Figures 4(a) and (b) present the corresponding simulated data in the same fashion.

Tables III and IV explain Figs. 5 to 17, which include only the cases for which the decision rule produced by the tracking algorithm did contain one or more switchings, because the no-switching cases are covered by one of the Figs. 3 and 4. The results, both for real and simulated data, did not present any major surprises. The choice of the reward matrix [c] and the multiplying factor of $-p(1-p)$ clearly influence

the decision-making process. For example, if we set the factor equal to zero and we penalize ($c = -1$) losing a target or going from state "U" of one target to state "D" of the other target, the algorithm decides no switchings. The same pattern is more pronounced if we have delays ($KD > 1$) because losing a target for one or more stages and thus increasing the corresponding penalties can be avoided by deciding no switchings. One way to change that no-switching decision is to penalize uncertainty, i.e., to have positive factors multiplying $-p(1-p)$ at each stage. Tables III and IV have the values of the factors used for Figs. 5–17. Another way to encourage switchings is to use the reward matrix (b) that does not penalize losing a target but instead rewards both up and down *crossings*, having zero reward for being in the "up" state. In the case of delays, however, unless our factor is nonzero, adopting reward matrix (b) will not produce a switching pattern; the decision will always be to listen to the same target instead.

It is also seen that adoption of the algorithms shown in this paper gives better results than listening to any one of the
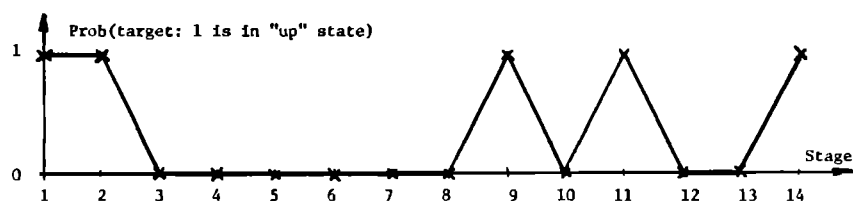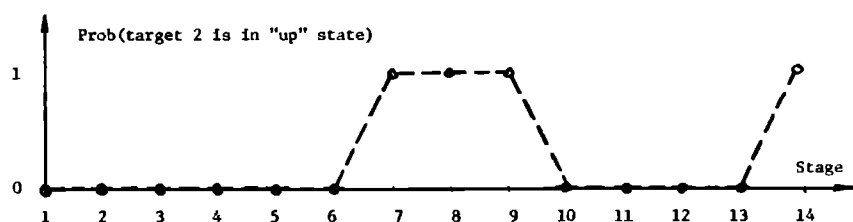

FIG. 4. Simulated data.

TABLE III. Real data.

Figure	5	6	7	8	9
Factor	1.0	1.0	4.0	2.0	2.0
Reward matrix	(a),(b)	(c)	(b)	(c)	(a)
KD	1	1	3	2	3

TABLE IV. Simulated data.

Figure	10	11	12	13	14	15	16	17
Factor	0.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0
Reward matrix	(a)	(a)	(a)	(a)	(b)	(b)	(b)	(c)
KD	1	1	2	3	1	2	3	1

FIG. 7.

FIG.5.

FIG. 8.

FIG. 6.

FIG. 9.

Prob(targets are in "up" state)

FIG. 10.

FIG. 11.

FIG. 12.

Prob(targets are in "up" state)

FIG. 13.

Fig. Franc

FIG. 14.

FIG. 15.

Prob(targets are in "up" state)
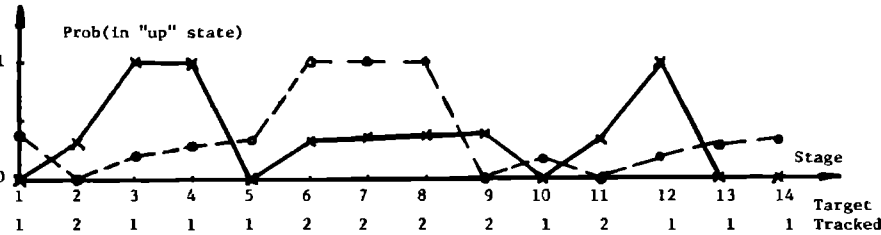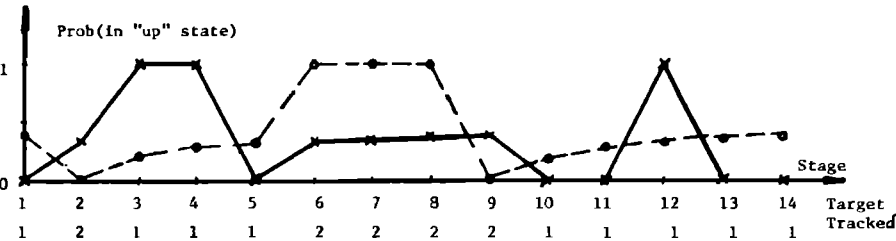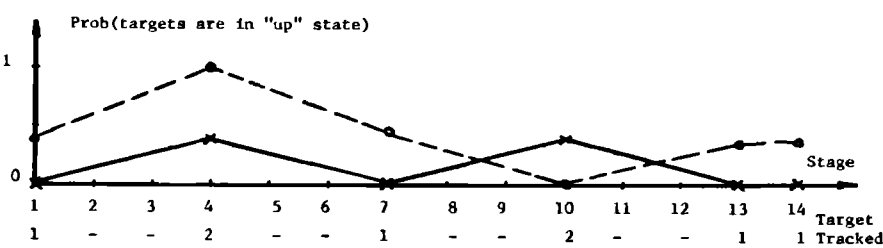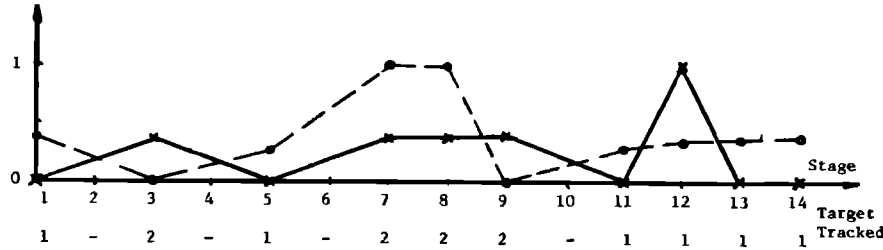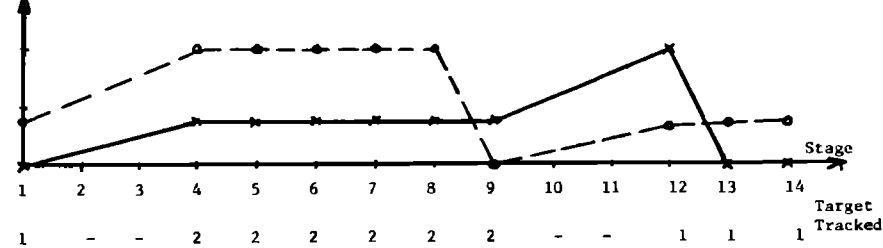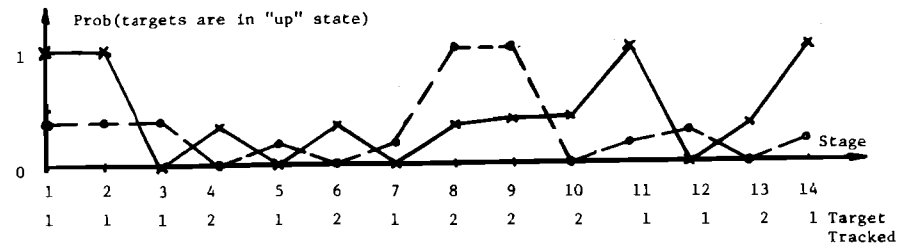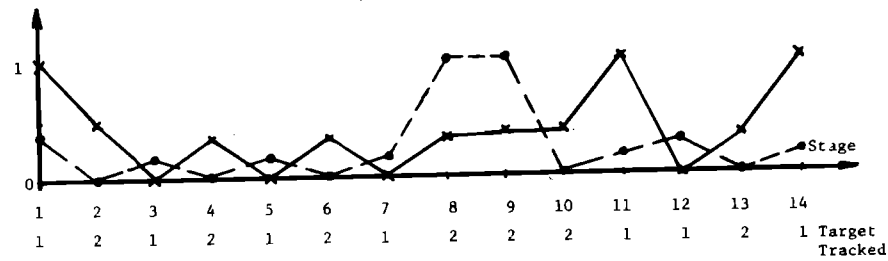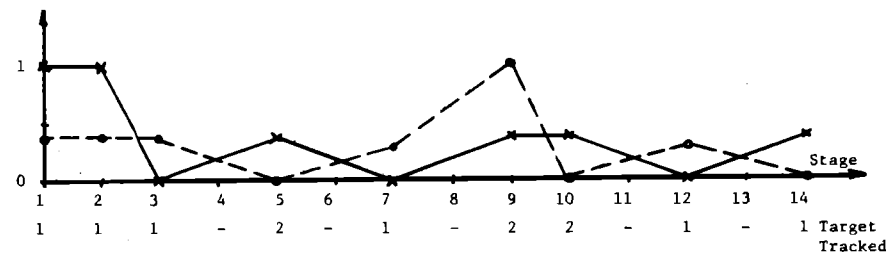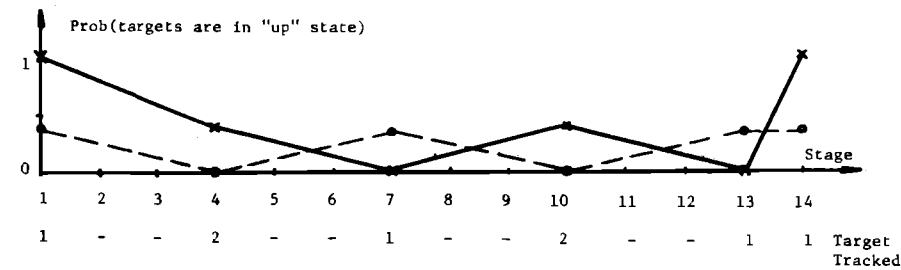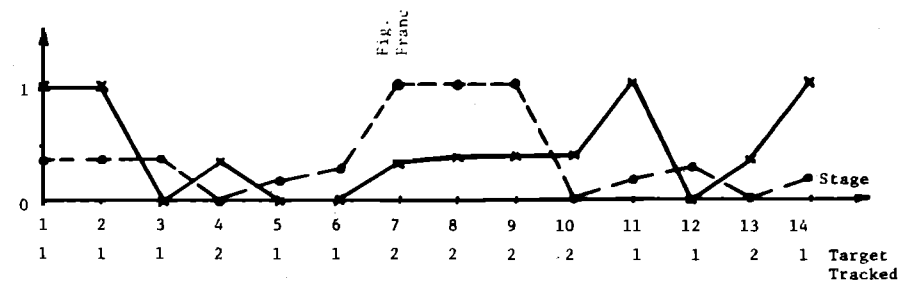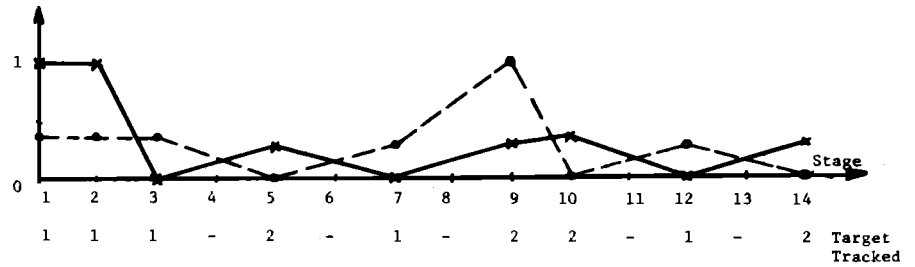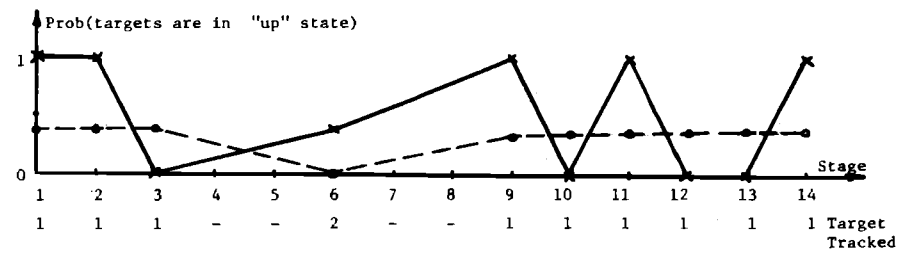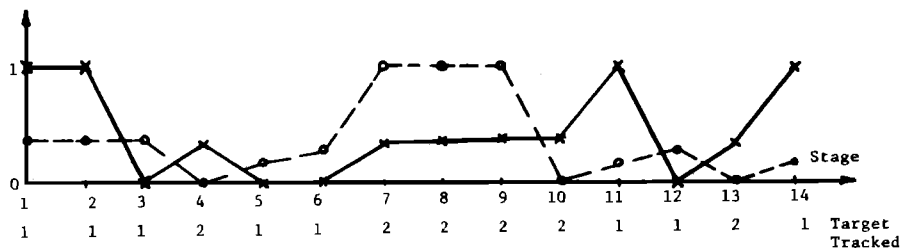
FIG. 16.

FIG. 17.

targets alone for the entire duration of the observations, in that the expected "reward-to-go" at each stage is always greater than or equal to the expected "reward-to-go" of the "nonswitching" policy for all cases examined. For example, for reward matrix (a), $KD = 1$ and factor $= 0$, the expected reward-to-go using our algorithms is 6.8910 at the beginning (stage one), whereas if we always listen to the same target our expected reward-to-go will be 5.446 and 5.194 for targets 1 and 2, respectively. Likewise, for the case of reward matrix (b) and factor $= 0$, $KD = 1$ we have an expected reward-to-go of 5.746 using the algorithm versus 5.58 and 3.304 if we listen to targets 1 and 2 (respectively) for the entire duration of the observations.

IV. CONCLUSIONS AND SUGGESTIONS FOR FURTHER RESEARCH

We have presented the formulation and solution of a simple resource allocation problem. The decision rules produced by the algorithms are heavily dependent on our choice of the reward and penalty terms as well as by the existence (or not) of delays in switching.

There are three basic limitations in the algorithms presented in this paper. First, only the computationally simplest case of one sensor and two targets is examined—usually we have several sensors and even more targets. Second, we assume stationarity of the detection process: the targets are or are not there for the whole duration of the observations. Third, noise is not treated here; in the real ocean, we can never be sure that $\rho \gg \rho_0$ implies that we indeed hold a target or that we just have excessive ambient noise. Hence, several directions of future research seem to be open at this point.

A. Extension to more sensors and/or targets

The problem of one sensor and several targets is a straightforward extension of the algorithms developed in this paper. The stage space will increase exponentially with the number of targets, and the available decisions will be equal in number to the number of targets. Consequently, for a large number of targets we should expect computational difficulties, which can be partially remedied by using heuristics to either approximate the decision regions or the objective function at each stage, or by grouping the targets appropriately and solve many smaller problems, getting in all cases suboptimal results. Before we proceed with such heuristics, however, it might be beneficial to obtain some exact results and see how they will be changed by our approximations.

The problem of two sensors and three targets can be considerably more complicated. In the simplest case, the sensors are all located in the same sonar array (i.e., they are individual sonar beams that can be steered in the direction of any target we choose to listen to); hence when two such sensors are tuned to the same target they receive the same signal. We have formulated dynamic programming algorithms for this case, with or without delays. The state space is three times as large as the one of this paper. The problem of $NS = 2$ sensors and $NT = NS + 1$ targets can be handled similarly. We have NT decisions (i.e., which target to leave out at each stage) and an $N \times NPR \times 2NS$ state space. Clearly, for many sensors we will have serious computational

problems, and we will probably have to resort to heuristics. The incorporation of delays in the above extensions is expected to be somewhat more complicated than in the problems formulated and solved in this paper. The case of sensors located in different parts of the ocean, i.e., each receiving a different signal from the same target, requires a doubling of the state space over the identical sensors case, but otherwise is not a conceptually more difficult problem.

The problem of $NS > 2$ sensors and $NT > NS + 1$ targets is the most general one. The number of available decisions increases significantly. Severe implementation problems are expected.

B. Extensions to nonstationary processes

Our models for the detection process hold, in the strict sense, for a stationary phase random multipath process. Clearly, if our targets appear and disappear relatively frequently, this stationarity is not preserved and our models cannot be used. However, under the assumption of infrequent changes (quasi-stationarity), we can accept that the process is characterized by one set of Markov models during the (long) interval before a change in the state of nature, and by another after the change (piecewise stationary processes). Such approaches have been already used in Ref. 14 in sequential hypothesis tests (failure detection). Their implementation within our target tracking framework is far from obvious at this point.

C. The treatment of noise

A straightforward way to take into account the "corruption" of our observations by noise would be to reformulate the modeling of our targets as partially observable Markov processes. This will significantly complicate both the formulation and the implementation of the problem, particularly for more than two targets. However, our present formulation might be interpreted in a way that satisfies this requirement, as follows: the detection threshold can be thought of as assigning our observations not to either the "up" or the "down" states presented in this and previous papers, but to either a "certain" ($\rho \gg \rho_0$) or an "uncertain" ($\rho < \rho_0$) state. This can be extended further by developing a three-state Markov model involving two thresholds and that will assign the observations in one of the following states: (a) target is there, $\rho \gg \rho_0'$ (b) target is not there, $\rho < \rho_0$, and (c) uncertain state, $\rho_0 \leqslant \rho < \rho_0'$. Developing this model is not expected to present serious problems.

Note added in proof: While this paper was with the printer, the authors have developed heuristic procedures for approximately solving the general multisensor, multitarget version of this problem. Details can be found in Ref. 17.

[1]D. L. Alspach, "A Gaussian Sum Approach to the Multi-Target Tracking Problem," Automatics II, 285–296 (1975).

[2]Y. Bar Shalom, "Application of Stochastic Control Theory to Resource Allocation Under Uncertainty," IEEE Trans. Autom. Control AC-19 (1) (1974).

[3]Y. Bar Shalom and G. D. Marcus, "Tracking with Measurements of Uncertain Origin and Random Arrival Times," Proc. 18th Conf. Decision Control, Ft. Lauderdale, FL (Dec. 1979).

[4]Y. Bar Shalom, "Tracking Methods in a Multitarget Environment," IEEE Trans. Autom. Control AC-23, 618–626 (1978).

[5]T. E. Fortmann and S. Baron, "Problems in Multi-Target Sonar Tracking," BBN Rep., N.E.S.C. # N00039-78-C-0296.

[6]T. E. Fortmann, Y. Bar Shalom, and M. Scheffe, "Multi-Target Tracking Using Joint Probabilistic Data Association," report for the CDC (1980).

[7]T. E. Fortmann, Y. Bar Shalom, and M. Scheffe, "Detection Thresholds for Multi-Target Tracking in Clutter," Rep. No. 4605, Bolt Beranek and Newman, Inc. (1981).

[8]B. Friedlander, "Multitarget Tracking Studies Phase I Final Report," Rep. No. 5334-01, Systems Control, Inc. (1980).

[9]K. M. Keverian and N. R. Sandell, "Multiobject Tracking by Adaptive Hypothesis Testing," LIDS-959, MIT laboratory for Information and Decision Systems (1979).

[10]D. Reid, "An Algorithm for Tracking Multiple Targets," IEEE Trans. Autom. Control AC-24, (12) (1979).

[11]H. N. Psaraftis, A. N. Perakis, and P. N. Mikhalevsky, "New Models on the Ocean Acoustic Detection Process," J. Acoust. Soc. Am. 69, 1724–1734 (1981).

[12]H. N. Psaraftis, A. N. Perakis, and P. N. Mikhalevsky, "Memory Detection Models for Phase-Random Ocean Acoustic Fluctuations," International Conference on Communications, Denver, CO (1981).

[13]J. J. Anton, "A Fluctuation Data Base for the CASE Experiment," Tech. Rep. S204-2, Systems Control, Inc. (1978).

[14]T. T. Chien and M. B. Adams, "A Sequential Failure Detection Technique and its Applications," IEEE Trans. Autom. Control. AC-21, 750–757 (1976).

[15]Y. Bar Shalom and E. Tse, "Tracking in a Cluttered Environment with Probabilistic Data Association," Automatica II, 451–460 (1975).

[16]Y. Bar Shalom and A. I. Cohen, "Optimal Resource Allocation for an Environmental Surveillance System," IEEE Trans. Syst., Man Cybernet. SMC-6 (6) (1976).

[17]H. N. Psaraftis and A. N. Perakis, "Myopic and Presbyopic Approaches to a Multi-Sensor, Multi-Target Resource Allocation Problem," 5th MIT/ONR Symposium on Command, Control and Communications, Monterey, CA (August 1982).